

# NOCIONES Y DESAFÍOS QUE INVOLUCRA EL DESARROLLO WEB.

Badwin Arévalo  
Ingeniero de Sistemas  
Docente catedrático Universidad Francisco de Paula Santander Ocaña

Sin duda la Web se ha convertido en una herramienta importante en nuestras vidas. La evolución desde las primeras páginas hasta las aplicaciones que usamos hoy en día ha sido vertiginosa, desde hace algunos años, poco a poco la Web 2.0 ha ido abriéndose camino en este mundo de la internet.

El lenguaje predominante en la elaboración de páginas Web es el HTML, sigla de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), el cual se ha convertido en un estándar y desde 1996 el organismo de estandarización llamado W3C World Wide Web Consortium (Consortio de la World Wide Web) es el encargado de la publicación de las recomendaciones para elaborar documentos HTML.

La Web como la conocemos en la actualidad, ha sufrido una transformación eminente, pasó de ser un simple documento de texto estático acompañado de imágenes, con poca interacción por parte del usuario y de escasa actualización, a convertirse en lo que comúnmente se conoce como la Web 2.0, etapa de la Internet en la que los usuarios son partícipes activos en la generación de contenidos debido a que éstos se crean dinámicamente, las actualizaciones se realizan en tiempo real y por lo general el uso de imágenes, audio, video, animaciones y todo tipo de interacciones es el factor común. Es

tanto el potencial de esta 'Web renovada', que la tendencia es que las aplicaciones orientadas a la Web sustituyan las aplicaciones de escritorio', como lo es el caso de Fixpicture<sup>2</sup>, que permite la edición de imágenes vía Web o el mismo Google Docs<sup>3</sup> que permite crear documentos de texto, hojas de cálculo o presentaciones a través de la Web.

El desarrollo Web implica muchas variables, pero ante todo debemos tener claridad en la diferencia que existe entre los lenguajes que se ejecutan del lado del cliente, los del lado del servidor, de los alcances que se tienen con cada uno de ellos y de la comunicación cliente-servidor.

Los lenguajes del lado del cliente son aquellos interpretados por el navegador Web<sup>4</sup>, el cual se encarga de analizar el documento recibido y de mostrarlo como una página Web.



Fig. 1. Iconos de algunos navegadores.



1. Aplicación de escritorio es aquella que requiere ser instalada para su uso y sólo puede ser utilizada en los equipos donde se hayan instalado.
2. Sitio Web para edición de imágenes en línea <http://www.fixpicture.org/>
3. Sitio Web para la generación de documentos en línea <http://docs.google.com/>
4. Entiéndase por navegador Web a la aplicación que se usa para navegar en internet como Internet Explorer, Mozilla Firefox entre otros.

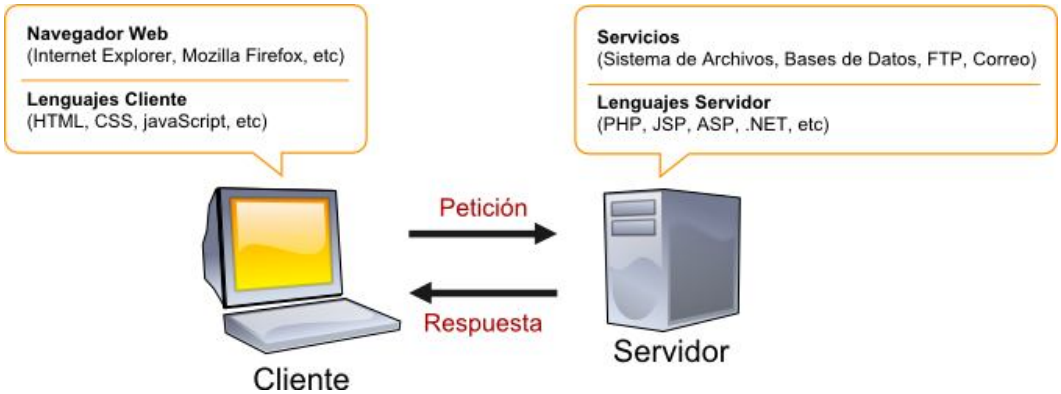


Fig. 2 Comunicación Cliente-Servidor

Entre los lenguajes más comunes del lado del cliente están, HTML, XHTML, Javascript, CSS entre otros.

Los lenguajes del lado del servidor, son aquellos que como su nombre lo indica, son interpretados por un servidor Web que se encarga de ejecutar las acciones indicadas por el programador y a su vez de generar y enviar un resultado 'traducido' a la aplicación cliente para que ésta sea capaz de mostrar los resultados obtenidos. Entre los lenguajes del lado del servidor más populares se encuentran: PHP, JSP, ASP, .NET, etc.

El uso de uno u otro depende de las acciones que se deseen ejecutar, por ejemplo, con los lenguajes del lado del cliente es posible crear diseños atractivos y funcionales, aplicar efectos, monitorear eventos, brindar interacciones, incorporar audio y video, validar formularios, etc.

Los lenguajes del lado del cliente tienen limitaciones, por ejemplo la dependencia a la capacidad de interpretación del navegador o la restricción para acceder a los recursos del sistema por motivos de seguridad, por lo que este tipo de lenguajes por sí solos no son suficientes para la creación de sitios Web completos y complejos.

Por otro lado los lenguajes del lado del servidor, permiten acceder a los recursos del servidor

remoto donde se halla alojada la página Web, lo que significaría acceso al sistema de archivos<sup>5</sup>, disponibilidad de servicios adicionales como motores de bases de datos, servidores FTP, librerías, servidores de correo, entre otros<sup>6</sup>.

En la figura 1, se muestra de manera muy simple los procesos que se llevan a cabo cuando por ejemplo se hace clic en un enlace o se envía información a través de un formulario. Éstos consisten en una petición por parte del cliente (Estos procesos), la cual lleva embebidos datos para que el servidor los analice y genere una respuesta, una vez recibida y aceptada la petición por parte del servidor éste determina qué acciones realizar, por ejemplo hacer uso del intérprete del lenguaje que se utilice en la aplicación, acceder a los servicios de un motor de bases de datos, servidor de correo o cualquier otro servicio, todo esto dependiendo de las instrucciones recibidas; una vez el servidor realice todas las tareas asignadas, genera una respuesta y la envía al cliente para que éste la interprete y la muestre al usuario.

Dentro del ámbito Web se pueden encontrar aplicaciones con diversas características que indudablemente influyen en la escogencia de las tecnologías, lenguajes y técnicas de programación a la hora de pensar en un desarrollo. Los que comúnmente se desarrollan en sitios personales, de empresas, ocio, educación, foros y demás, son

5. Esto es posible si se han asignado los permisos respectivos en el servidor.

6. Los servicios adicionales son accesibles siempre y cuando estén instalados en el servidor y se haya concedido un nombre de usuario y una contraseña para su uso.

los que incluyen los lenguajes cliente y servidor además de un motor de bases de datos para almacenar información.

Existen también los que son denominados RIA Rich Internet Applications (Aplicaciones de Internet Enriquecidas) que son las que buscan integrar características de las aplicaciones de escritorio tales como menús contextuales, barra de tareas, barra de herramientas etc. Para lograr la funcionalidad de estas características añadidas en ocasiones es necesario que el usuario instale un plugin, como es el caso de Flash, Silverlight, Openlaszlo o Unity por mencionar algunos.

Algunas aplicaciones son conocidas como Aplicaciones Web Híbridas o Mashups, éstas consisten en la integración y uso de servicios ofrecidos por terceros mediante un API Application Programming Interface (Interfaz de Programación de Aplicaciones) público que permite invocar los servicios de una aplicación en otra. Algunos ejemplos de servicios con APIs disponibles en la actualidad son: Google Maps, Flickr y Youtube.

Por último se sacan a colación los Web Services (Servicios Web) que son aplicaciones que trabajan de manera colaborativa y comparten datos a pesar de estar escritas en lenguajes distintos, esto lo logran por medio de protocolos y estándares definidos.

Ahora estando en contexto y con el conocimiento algunos de los diversos tipos de aplicaciones que pueden existir falta claridad no entiendo? la atención se centra en aquellos puntos que se deben considerar cada vez que se asume la tarea de desarrollar de una aplicación Web (Langridges, 2005). Ante todo se debe tener claro qué tipo de aplicación es a la que se quiere llegar, cuál es la población a la que va dirigida, y cuánto conocimiento previo deben tener los potenciales usuarios. Lo anterior es debido que al tener en cuenta estas consideraciones básicas se puede entrar a discutir y definir por ejemplo qué lenguajes escoger según el tipo de aplicación o si el usuario debe tener algunos conocimientos intermedios para realizar tareas tales como instalar un plugin

en caso que la aplicación lo requiera. No dejar a un lado estos y otros cuestionamientos pueden influir en el éxito o fracaso de una aplicación, como en el caso de los conocimientos previos que el usuario deba tener, porque si se desarrolla una aplicación pública que estará disponible para infinidad de usuarios y ésta requiere de algunos conocimientos intermedios, lo más probable es que la mayoría de los usuarios inexpertos opten por abandonar la aplicación porque ésta no está a su alcance, por lo que tener en claro la población objetivo se convierte en factor clave, ya que se supone que la finalidad es llegar a la mayor cantidad de usuarios posibles.

En la actualidad el desarrollo de una aplicación Web por lo general va ligado a un motor de bases de datos, en el mercado existen muchos de ellos, cada uno con características específicas, las cuales el desarrollador decide cuáles de ellas son las más pertinentes para la aplicación. Algunos de los motores existentes son: MySQL, PostgreSQL, Oracle, SQL Server o Firebird. Aunque uno de los más populares en aplicaciones Web es MySQL.

Sin duda la gran mayoría de las aplicaciones Web van de la mano de los navegadores, ellos juegan un papel supremamente importante y lógicamente cuando se desarrolla hay que tenerlos muy en cuenta (Zantdstra, 2008). En el mercado existen gran variedad de navegadores Web, cada uno con características y funcionalidades que los diferencian, de los cuales los usuarios hacen uso de uno o más de ellos según sus preferencias. Entre los navegadores más populares se encuentran: Internet Explorer, Mozilla Firefox, Opera, Google Chrome, Safari, Konqueror, entre otros. La cuota de mercado, es decir la preferencia de uso de un navegador u otro según la compañía Net Applications, para Marzo de 2010 es la siguiente:

Microsoft Internet Explorer 60.65%  
Mozilla Firefox 24.52%  
Google Chrome 6.13%  
Safari 4.65%  
Opera 2.37%

7. [http://marketshare.hits.ink.com/browser\\_market\\_share.aspx?qpr'd\\_0](http://marketshare.hits.ink.com/browser_market_share.aspx?qpr'd_0)



Como se puede observar el 98.32% de la preferencia de uso de los navegadores está conformada por 5 de ellos, aunque el total de navegadores es relativo porque infortunadamente las distintas versiones de Internet Explorer interpretan de manera distinta las páginas Web, es decir con este navegador es necesario tener en cuenta sus versiones 6, 7, 8 y próximamente será lanzada la versión 9.

Uno de los errores que comúnmente se encuentran en el desarrollo de aplicaciones Web, sobre todo en principiantes en este medio, es no tener en cuenta las diferencias que existen entre los distintos navegadores, generalmente (omisión de por lo que) desarrollan para que la aplicación funcione a la perfección con Internet Explorer y peor aún sólo para una versión en especial; esto en años anteriores no era considerada una mala práctica, puesto que existían menos versiones de Internet Explorer y además éste contaba con algo más del 90% de la cuota del mercado, pero hoy en día esto significa marginar a aproximadamente el 40% de los potenciales usuarios.

Lograr que una aplicación Web sea Crossbrowser, es decir que ésta sea compatible con los distintos navegadores debe ser uno de los principales objetivos en el desarrollo, garantizando de esta manera cobijar la mayor cantidad de usuarios posibles. Al hablar de Crossbrowser principalmente se involucran dos lenguajes: CSS y Javascript. Aunque la mayoría de navegadores modernos soportan los estándares CSS emitidos por la W3C, el caso específico de Internet Explorer (en especial de la versión 7 hacia atrás) se ha convertido en un verdadero dolor de cabeza para los desarrolladores, debido a que estas versiones del navegador no interpretan de manera adecuada ciertas propiedades CSS, no soportan algunas pseudoclases en todos los elementos y son muy limitantes al momento de utilizar selectores, lo que en muchas ocasiones obliga a la utilización de hacks (recortes, parches) para que el comportamiento del CSS sea el esperado, provocando que el código resulte mucho más extenso.

En el caso de Javascript el principal inconveniente

es que algunas de las funciones nativas del lenguaje no están disponibles en todos navegadores o toman nombres distintos para realizar una acción determinada, este condicionamiento es particularmente tedioso a la hora de buscar que nuestra aplicación sea Crossbrowser, debido a que hace falta encontrar la manera para que la aplicación se comporte de forma similar en todos los navegadores. Esto se logra encontrando funciones equivalentes entre los distintos navegadores o creando unas propias que realicen la tarea esperada sin importar el navegador. En la actualidad el tema de compatibilidad de Javascript se puede solucionar utilizando Frameworks o librerías, los cuales poseen funciones que son totalmente compatibles entre navegadores, lo que de alguna manera facilita y agiliza el trabajo, aunque hay que tener en cuenta el tiempo que toma conocer y aprender a utilizar estas funciones creadas por terceros.

Los grandes inconvenientes de apariencia y comportamiento que generan navegadores obsoletos como el Internet Explorer 6, ha incentivado campañas en La Internet como el NOIE6 Day (Día del no al Internet Explorer 6) o la creación de sitios que promueven la no utilización de este navegador como es el caso de [www.ie6nomore.com](http://www.ie6nomore.com), donde se incentiva a insertar un código en la aplicación que muestra un anuncio al usuario para que cambie o actualice su navegador; una campaña similar la lidera el portal de videos Youtube donde se anuncia al usuario que el navegador no es compatible y que debe ser cambiado o actualizado.

La adopción de ciertas prácticas puede mejorar el proceso de desarrollo de una aplicación Web, una de las que se debe considerar es la de separar los contenidos (HTML, XHTML), el diseño (CSS) y la lógica (programación, Javascript u otro lenguaje), es decir no mezclarlas porque de esta manera lo que se logra es "ensuciar" (comilla simples se usan luego de usar comillas dobles, en este párrafo solo has usado una vez comillas) el código al tener todo esto en un único archivo o lugar. Para lograr esta separación es aconsejable crear un documento

8. Porta para a publicación y visualización de videos en línea. <http://www.youtube.com/>

independiente para cada una de las partes (contenido, diseño y lógica), donde por medio de inclusiones o llamados desde un documento por lo general HTML, se puede tener disposición de los otros (CSS y Javascript).

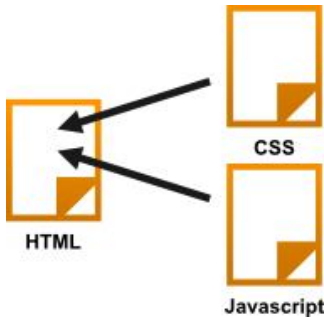


Fig.3 inclusión de documentos externos

Esta práctica evita la duplicación de código ya que permite ser incluido en cualquier documento donde sea necesario sin tener que reescribir ninguna línea, además brinda otra ventaja importante y es el fácil mantenimiento, porque al estar separadas cada una de las partes, se pueden hacer modificaciones al código pertinente sin realizar búsquedas engorrosas en archivos de gran tamaño, además los cambios se ven reflejados en todos y cada uno de los documentos donde es incluido el archivo modificado.

## Conclusiones

Es aconsejable intentar en lo posible respetar los estándares, con esto se incrementan las posibilidades de que la aplicación sea Crossbrowser, aunque en algunas situaciones se tenga que recurrir a prácticas no recomendadas.

Cada vez que se haga un avance importante en el desarrollo de la aplicación, ésta debe ser testeada con los diversos navegadores, así conforme se vaya creando se asegura el correcto funcionamiento de la misma, y se pueda detectar de manera temprana los posibles errores o incompatibilidades existentes.